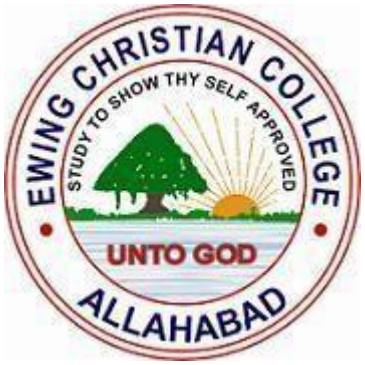
Bachelor of Computer Application

PROJECT REPORT

Business Management System (Workfusion)

By
Asmita Singh
(ECC2314032)
(25413027)

Adviser:
Mr. Abhishek Srivastava
DEPARTMENT OF Computer Application



Ewing Christian College, Prayagraj

Gaughat, Prayagraj Uttar Pradesh: 211003

November , 2025

Project Evaluation Sheet

Sr. No	Mini Project	Marks
1	Record [60]	
2	Demonstration [60]	
3	Viva [60]	
	Total [180]	

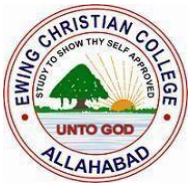
Examiner-1 Examiner-2

Declaration

I declare this written submission represents my ideas in my own words, and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misinterpreted or fabricated or falsified any idea/data/fact/source in my submission. I understood that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

1. Asmita Singh **ECC2314032** **Signature**

Date: _____



CERTIFICATE

This is to certify that the project entitled "Bussiness Management System (WORKFUSION)" is the bonafide work carried out by **Asmita Singh**, student of BCA Semester V, Ewing Christian College, Prayagraj during the year 2025, in partial fulfilment of the requirements for the award of the Degree of Bachelors of Computer Application.

Advisor Mr. Abhishek Srivastava

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to **Mr. Abhishek Kumar Srivastava,** my esteemed advisor, whose unwavering support and expert guidance have been invaluable in completing my project, **"Bussiness Management System (WORKFUSION)."**

His mentorship and patience have been a source of continuous encouragement, and his insights and dedication have greatly enriched both the project and my personal learning journey. This work would not have been possible without his wisdom and direction.

I am also profoundly thankful to my family, friends, and colleagues for their constant support and understanding. Their encouragement helped me push through challenges and stay focused, and their belief in my potential has been a source of immense motivation.

Finally, I would like to extend my appreciation to everyone who, directly or indirectly, contributed to this project. This experience has taught me so much, and I look forward to carrying these lessons forward with a renewed sense of commitment and passion.

ABSTRACT

The project titled **"Business Management System (WORKFUSION)"** has been developed to automate and simplify the various business operations of an organization through a centralized web-based platform. In the modern corporate environment, businesses often struggle with managing employee data, attendance, payroll, leave tracking, meeting scheduling, and internal communications efficiently.

The system has been designed to streamline organizational workflows, offering two main user roles — **Administrator** and **Employee**. The **Administrator** can manage employee records. The **Employee** can mark attendance, apply for leaves, check payroll details, access assigned tasks, and stay updated with company notices. This ensures transparency, accountability, and consistent communication between staff and management. The application has been implemented using open-source web technologies — **PHP** for serverside scripting, **MySQL** for database management, **HTML** and **CSS** for interface design, and **JavaScript** for dynamic interactivity.

Table of Contents

1. INTRODUCTION

The Business Management System (WORKFUSION) is a comprehensive web-based platform developed to automate and manage core business activities. The system allows administrators to handle employee records, attendance, leave approvals, salary generation, meeting schedules, and project announcements all in one unified portal. In today's digital world, manual business operations can be time-consuming and error-prone. WORKFUSION aims to eliminate these issues by providing a digital interface where business data is stored, processed, and monitored in real time.

The primary aim of the system is to provide both administrators and employees with an accessible, secure, and easy-to-use environment where all organizational tasks can be monitored and executed effectively. The administrator can manage employee details, approve leaves, calculate payroll automatically, assign projects, and post announcements — while employees can mark attendance, apply for leaves, view salary slips, and stay informed about company updates. This two-way communication system enhances organizational discipline and helps in maintaining transparency across departments.

From a technical perspective, WORKFUSION has been built using open-source technologies — **PHP** as the backend programming language, **MySQL** for database management, **HTML** and **CSS** for front-end design, and **JavaScript** for client-side interactivity. These technologies were chosen for their reliability, scalability, and ease of deployment in realworld business environments. The system operates efficiently on both local and online servers, ensuring data consistency and quick access for all authorized users.

2. OBJECTIVES

The main objective of developing the **Business Management System (WORKFUSION)** is to create an integrated, reliable, and automated platform that simplifies the management of all essential business activities in a digital environment.

Centralized Business Operations:
To integrate multiple business functions — such as employee records, attendance, leave, payroll, and announcements — within a single web-based system for efficient management.

Employee Management:
To maintain accurate employee information, including department, designation, and salary details, in a structured digital format.

Automated Attendance Tracking:
To allow employees to mark attendance online, with the system automatically marking absentees after a specific time limit.

Leave Management with Limits:
To simplify leave applications while enforcing yearly and monthly leave limits and providing alerts when these limits are exceeded.

Payroll Automation:
To generate monthly salary slips automatically based on attendance, leave, and deductions, ensuring accuracy and transparency.

Meeting Scheduling:
To enable administrators to schedule and manage meetings efficiently and notify employees of upcoming sessions.

Deduction & Financial Summary:
To maintain transparent records of deductions for attendance or excess leave and present them clearly in employee dashboards.

Task & Project Assignment:
To assign work or projects digitally, set deadlines, and allow progress tracking and reporting.

Announcements & Notices:
To provide an internal platform for posting company news, policy updates, and event notifications.

Security & Scalability:
To ensure secure user authentication, data protection, and system scalability for future expansion such as mobile integration.

3.PROJECT SCOPE

The **Business Management System (WORKFUSION)** has been developed with the objective of providing a complete and unified digital solution for managing an organization's core business activities. The project covers a wide range of administrative and operational functions such as employee management, attendance tracking, leave processing, payroll generation, meeting scheduling, task assignment, and internal announcements. The system serves as a centralized web-based platform that brings together all essential modules required for the smooth functioning of a business, thereby minimizing manual work and increasing overall efficiency.

The scope of this project includes the design and implementation of a secure, user-friendly, and scalable management system that can be accessed by both administrators and employees. Administrators have complete control over the platform, allowing them to register and manage employee details, monitor attendance, approve or reject leave applications, calculate payrolls automatically, track deductions, schedule meetings, assign tasks, and post company announcements. On the other hand, employees can log in to mark attendance, apply for leave within their allowed limits, view salary slips, check task assignments, and stay informed about official news and notices through their dashboard. This dual-level access structure ensures transparency, accountability, and smooth communication within the organization.

The project's scope extends beyond simply automating day-to-day business operations; it focuses on improving coordination, accuracy, and productivity across all departments. By shifting from manual to automated processes, organizations can save time, reduce errors, and maintain consistent records of all business activities. In addition, the system promotes transparency between management and employees by making all necessary information — such as leave balances, salary breakdowns, and assigned projects — accessible in real time.

4.PROJECT FEATURES

The proposed Business Management System is designed to streamline and automate essential organizational activities such as employee management, attendance monitoring, payroll processing, leave handling, and meeting scheduling. The system ensures efficiency, transparency, and accuracy in managing daily operations. Below are the key modules and their detailed functionalities.

- **Employee Management:**

  The **Employee Management** module serves as the foundation of the system. It allows administrators to **register, update, and manage employee details** such as name, employee ID, department, designation, contact information, and joining date.

  Employees can securely **log in to the system** to access their personal information, attendance records, and payroll history. This centralized database eliminates the need for manual record-keeping and ensures data consistency across all modules.

  Furthermore, this module supports **role-based access control**, ensuring that administrators, HR personnel, and employees only access functionalities relevant to their responsibilities. It is tightly integrated with other system modules like Attendance, Payroll, and Leave Management to maintain accurate employee data across all operations.

- **Attendance Tracking**

  The **Attendance Tracking** module provides an automated and reliable way of recording employee presence. Employees can **mark their attendance by checking in and checking out** through the system. The system records timestamps and calculates total working hours automatically.

  A built-in **automated attendance verification script** marks absentees if they fail to check in before a predefined cutoff time (for example, 11:00 AM). This automation minimizes manual errors and ensures that attendance data remains accurate and uptodate.

  Administrators can generate **attendance reports** for specific employees, departments, or date ranges, which are then used in payroll and performance evaluations.

- **Leave Management with Limit**

  The **Leave Management** module simplifies the process of applying for, approving, and tracking leaves. Employees can submit leave applications for various categories such

as **Casual Leave (CL), Sick Leave (SL), Maternity Leave (ML) and Paid Leave (PL)** directly through the system interface.

To ensure fair and transparent leave distribution, the system enforces **monthly and yearly leave limits**—for instance, 10 CLs and 3 SLs per month. If an employee exceeds these limits, the system automatically **generates an alert or warning message**, while still recording the request for administrative review.

Administrators can review, approve, or reject leave requests with a single click. The system updates leave balances in real time, ensuring accurate tracking and preventing misuse of leave privileges.

- **Payroll Management**

The **Payroll Management** module automates salary processing based on attendance, approved leaves, and applicable deductions. It ensures error-free calculations, saving time and effort for HR and accounting teams.

Each employee's **salary components**—including basic pay, house rent allowance (HRA), travel allowance, and other benefits—can be customized by the administrator. The system then automatically computes **gross salary, deductions, and net pay** based on attendance and leave data.

At the end of each month, a detailed **payroll summary report** is generated, showing total earnings, deductions, taxes, and final payable amounts. This enhances payroll transparency and helps employees understand how their salaries are computed.

- **Meeting Scheduling**

The **Meeting Scheduling** feature enables administrators to **plan, schedule, and manage meetings** efficiently. Meetings can be organized for specific departments, teams, or individual employees.

Once a meeting is created, all concerned participants receive **notifications through their dashboards**. The system stores meeting details, including agenda, date, time, and participants, for future reference.

Additionally, admins can **track meeting attendance** and generate participation reports, ensuring accountability and improving coordination among teams.

- **Deduction Summary**

  The **Deduction Summary** module provides employees with a **transparent breakdown of salary deductions**. It displays all adjustments made due to **leaves without pay, late attendance, tax deductions, or other penalties**.

  This feature ensures that employees are fully informed about how their net salary is computed. For administrators, it serves as a quick reference to verify payroll accuracy and maintain financial transparency within the organization.

- **Task Assignment**

  The **Task Assignment** module allows administrators or managers to assign **tasks, projects, or goals** to specific employees or teams. Each task includes details such as description, start date, deadline, and assigned personnel.

  Employees can **update task progress, add remarks, and mark completion** once finished. This real-time task tracking ensures accountability and enables administrators to monitor project timelines effectively.

- **Project Announcements / News & Notices**

  The **Announcements and Notices** module functions as an **internal communication board** for the organization. It allows administrators to post important **announcements, company news, upcoming events, or urgent notifications**.

  Employees can view all notices directly on their dashboards upon login. This promotes better communication and ensures that everyone stays informed about organizational updates and policy changes.

  The announcement module also supports **categorized posts** (e.g., “Policy Updates,” “Events,” or “General Notices”) and includes timestamped entries for easy reference and archiving.

5. TECHNOLOGIES USED

The Business Management System (WORKFUSION) has been developed using a combination of open-source web technologies that provide reliability, scalability, and efficient data handling. The system follows a three-tier architecture consisting of the frontend, backend, and database layers, each performing specific functions essential for the smooth operation of the platform.

- **Frontend Technologies**

  The front end of the system is developed using HTML, CSS, and JavaScript.

  HTML (Hypertext Markup Language) provides the structural layout of the web pages and defines all visual components such as forms, tables, and navigation menus.

  CSS (Cascading Style Sheets) is used to enhance the appearance and responsiveness of the interface. It ensures a clean, professional design that adapts to different devices like desktops, tablets, and mobile phones.

  JavaScript adds interactivity and real-time functionality to the web pages. It validates input fields, updates content dynamically and improves user experience.
  Additionally, libraries such as jQuery and AJAX are used to make the interface more responsive and to enable asynchronous communication between the client and server without reloading the entire page.

- **Backend Technologies**

  The backend of WORKFUSION is developed using PHP (Hypertext Preprocessor). PHP is an open-source, server-side scripting language that handles the business logic of the application.
  The application is developed and tested locally using the XAMPP Server, which integrates Apache, PHP, and MySQL in a single environment. This setup allows for smooth execution, debugging, and deployment of the project.

- **Database Management**

  The database layer of the system is managed using MySQL, a powerful and scalable relational database management system (RDBMS). MySQL is responsible for storing and organizing all essential business data, including employee details, attendance logs,

leave applications, salary records, and company announcements. It ensures data integrity, security, and quick retrieval through structured queries.

The system maintains relationships between multiple tables using primary and foreign keys, enabling efficient data management.

6.FEASIBILITY STUDY

Before developing any software project, it is essential to conduct a feasibility study to determine whether the project is practical, achievable, and beneficial in terms of technology, cost, and implementation. The feasibility study for the Business Management System (WORKFUSION) evaluates the technical, operational, economic, and schedule aspects of the system to ensure that the project can be successfully developed and deployed within the given constraints and resources.

- **Technical Feasibility**

  The technical feasibility of the WORKFUSION project is highly positive. The system is built using open-source technologies such as PHP, MySQL, HTML, CSS, and JavaScript, which are readily available, well-documented, and supported across different operating systems. These technologies are reliable, lightweight, and compatible with standard hardware and software configurations, making the system easy to implement and maintain. The use of XAMPP Server simplifies local testing by integrating Apache, PHP, and MySQL into a single environment. Furthermore, the development team possesses sufficient knowledge and experience in these technologies, ensuring smooth coding, integration, and debugging throughout the development process. Hence, from a technical perspective, the project is feasible and sustainable with minimal resource requirements.

- **Operational Feasibility**

  Operational feasibility examines whether the system will function effectively within the organization and meet user expectations. The WORKFUSION system automates essential business processes such as employee management, attendance tracking, leave approvals, payroll computation, meeting scheduling, and notice publishing — all through a single centralized platform. This significantly reduces paperwork, manual errors, and delays in communication. The user interface has been designed to be simple, intuitive, and accessible to users with minimal technical knowledge. Both administrators and employees can easily perform their respective operations with clear navigation and real-time feedback. The system enhances efficiency, transparency, and collaboration among employees, which contributes to better organizational productivity. Therefore, operationally, the project is highly feasible and will bring considerable improvements to day-to-day business management.

- **Economic Feasibility**

  Economic feasibility evaluates whether the project's benefits outweigh its costs. Since the Business Management System (WORKFUSION) is developed using open-source technologies, there are no licensing or subscription costs involved. The development requires only basic computing resources and a local server setup, keeping expenses low. The long-term financial benefits of the system include reduced administrative costs, minimized human errors, and time savings due to automation. Additionally, the project improves employee efficiency and reduces the dependency on manual labor for data management and payroll processing.

- **Schedule Feasibility**

  Schedule feasibility focuses on whether the project can be completed within the planned timeframe. The development of WORKFUSION followed an iterative model, which divides the work into smaller, manageable phases such as planning, designing, coding, testing, and implementation. Each module — including attendance tracking, leave management, and payroll — was developed and tested in successive iterations, allowing early detection of issues and timely corrections. The structured timeline ensured that the project progressed steadily and met its milestones as planned. Proper time management, task distribution, and regular progress reviews contributed to completing the project within the scheduled duration. Thus, the project is schedulefeasible and aligns well with the academic and practical development timelines.

- **Social and User Feasibility**

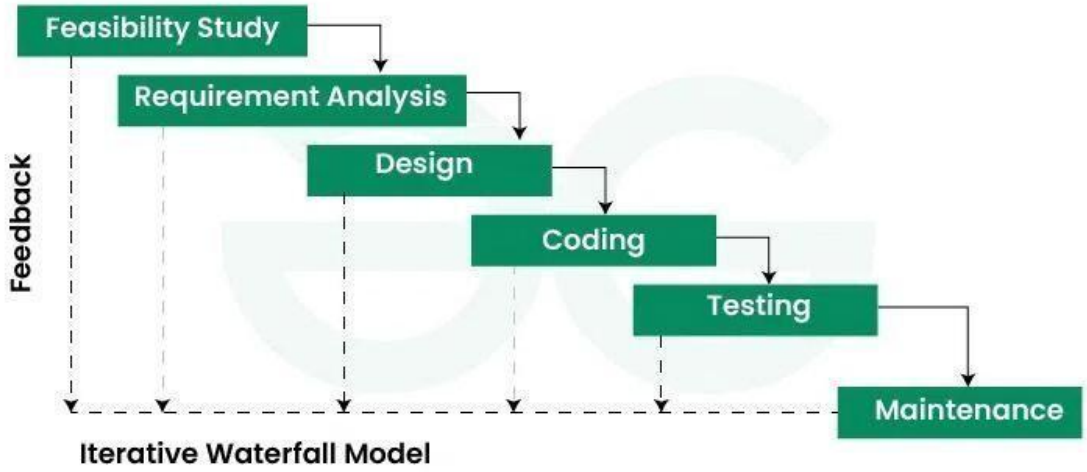
  From a social and user perspective, the system has been designed keeping in mind the needs of both administrators and employees. It provides a transparent environment that builds trust by making data such as attendance, salary, and leave records easily accessible. This fosters a sense of accountability and discipline among employees. The system also minimizes the administrative workload, allowing management to focus on strategic business decisions rather than repetitive manual tasks. Hence, it is socially beneficial and user-friendly, ensuring high acceptance among its intended users.

7.SDLC MODEL

The Software Development Life Cycle (SDLC) is a structured process that outlines the steps required to design, develop, test, and deploy software efficiently. It ensures that software products meet user expectations, are delivered on time, and function reliably within their intended environment. For the Business Management System (WORKFUSION), the Iterative Model of SDLC was adopted because it allows for continuous improvement through multiple development cycles, making it ideal for a modular and evolving system like this one.

ITERATIVE MODEL

The Iterative Model is based on the concept of building software in small portions or iterations. Instead of delivering the entire product at once, the system is developed incrementally — each iteration adds new features and refines the existing ones based on testing and user feedback. This model provides flexibility to accommodate changes during development and helps in early detection of errors, ensuring a more stable and user-friendly final product.



- **Key Characteristics of the Iterative Model**

The Iterative Model divides the entire development process into smaller, manageable cycles, each consisting of planning, analysis, design, implementation, and testing. At the end of each iteration, a working prototype of the system is produced. This approach ensures that any flaws

or improvements can be identified and incorporated into subsequent iterations. The main characteristics include:

1. Continuous user feedback after each iteration.
2. Incremental enhancement of features and functionalities.
3. Reduced project risk through early testing and validation.
4. Flexibility to adapt to changing requirements during development.

• Phases of the Iterative Model

The development of the WORKFUSION project followed these phases during each iteration:

1. Requirement Analysis and Planning:
In the initial phase, project requirements were collected through observation and analysis of existing business processes. The main objective was to understand the need for automation in attendance tracking, leave management, payroll generation, and internal communication. A detailed project plan was prepared, defining system goals, development tools, timelines, and user roles.

2. System Design:
Based on the gathered requirements, the system's architecture was designed to ensure modularity and scalability. The design phase included database schema creation using MySQL, user interface design with HTML and CSS, and planning the backend logic with PHP. Data flow diagrams (DFDs) were also designed to visualize interactions between different modules like employee management, leave tracking, and payroll.

3. Implementation (Coding):
After each iteration, the newly added features were thoroughly tested to ensure they worked correctly and did not affect previously developed modules. Unit testing, integration testing, and user acceptance testing were conducted. Any errors or performance issues detected during testing were resolved before proceeding to the next iteration.

4. Testing and Debugging:
After each iteration, the newly added features were thoroughly tested to ensure they worked correctly and did not affect previously developed modules. Unit testing, integration testing, and user acceptance testing were conducted. Any errors or performance issues detected during testing were resolved before proceeding to the next iteration.

5. Evaluation and Feedback:
Each prototype of the system was reviewed by users (admin and employee testers) to gather feedback on functionality, design, and usability. This feedback helped the developers make necessary modifications and improvements in subsequent iterations. For example, alerts for exceeding leave limits and real-time payroll summaries were added after feedback in earlier testing cycles.

6. Integration and Deployment:
Once all modules were developed and tested individually, they were integrated to form the complete system. The final deployment was carried out on a local XAMPP server, making the system ready for practical use. It can also be easily deployed on a live web server for realworld applications.

8.REQUIREMENT GATHERING AND ANALYSIS

The requirement gathering and analysis phase is one of the most critical stages in the development lifecycle of the **Business Management System (WORKFUSION)**. This stage serves as the foundation upon which the entire project is built. The main purpose of this phase is to clearly identify and document the requirements of the users, analyze their needs, and translate them into detailed functional and non-functional specifications that guide system design and implementation.

Requirement Gathering Process

To develop WORKFUSION effectively, information was gathered using several systematic approaches. The methods included:

1. **Observation of Existing System:** The current manual process of managing business operations such as employee attendance, leave management, payroll calculation, and project coordination was closely studied. This helped in identifying inefficiencies, data redundancy, and time delays caused by manual record-keeping.

2. **Interviews and Discussions:** Direct discussions were conducted with employees, team leaders, and administrators to understand their daily workflow, pain points, and expectations from the new system. These sessions provided valuable insight into the challenges faced, such as difficulty in tracking attendance, managing leave balances, and generating reports manually.

3. **Document Analysis:** Existing documents such as attendance registers, salary slips, leave application forms, and project schedules were analysed to understand the data flow, dependencies, and reporting structure. This helped in determining what information needed to be digitized and automated.

Feasibility Study:

A feasibility analysis was conducted to evaluate the technical, operational, and economic aspects of developing the system using **HTML, CSS, JavaScript, PHP, and MySQL**. The chosen technologies were found to be cost-effective, reliable, and scalable for the intended organization size. Through this combination of methods, clear and measurable requirements were established for WORKFUSION.

Requirement Analysis

Once the data was collected, it was analysed to determine **user expectations**, **system limitations**, and **business rules** that the new system must follow. The analysis identified several challenges in the existing manual process, such as:

- Lack of centralized data storage, leading to duplication and inconsistency.
- Difficulty in monitoring employee attendance and leave records.
- Manual calculation of payroll, which was time-consuming and prone to errors.
- Poor coordination between management and employees regarding meetings and announcements.
- Limited access to real-time reports and analytics.

The requirement analysis helped define a system that automates these processes and improves efficiency, transparency, and accuracy across the organization.

1. Functional Requirements

Functional requirements describe **what the system should do**—the core features and operations that must be implemented to satisfy user needs. For the **WORKFUSION** system, the following functional requirements were identified:

1. **User Authentication and Access Control:**

   The system should provide secure login functionality for both administrators and employees using unique credentials.

   User roles should determine access levels (e.g., employees can apply for leave, while administrators can approve/reject requests).

2. **Employee Management:**

   The system should allow administrators to add, edit, view, and delete employee records. Employee profiles should store personal information, contact details, and employment history.

3. **Attendance Management:**

   Employees should be able to mark attendance daily, either through a web interface or automated time tracking.

   The system should record attendance data and allow administrators to generate monthly attendance reports.

4. **Leave Management:**

   Employees should be able to apply for leave by specifying leave type, start and end date, and reason.

The system should automatically validate available leave balance before approval.

Administrators should have options to approve or reject leave requests and view overall leave statistics.

5. **Payroll Management:**
   The system should automatically calculate employee salaries based on attendance, leaves, and deductions.

   It should generate downloadable salary slips for each employee.

6. **Meeting and Task Scheduling:**
   Administrators should be able to create and manage meetings and assign tasks to specific employees or teams.

   Employees should be notified of scheduled meetings and tasks through the system interface.

7. **Announcements and Notifications:**
   The system should allow administrators to post company-wide announcements visible on employee dashboards.

   Notifications should be triggered for important actions such as leave approvals, new tasks, or payroll updates.

8. **Report Generation:**
   The system must generate analytical reports such as attendance summaries, leave records, and payroll statements.

   Reports should be exportable in common formats like PDF or Excel for administrative purposes.

9. **Database Management:**
   All data should be stored in a centralized **MySQL** database to ensure consistency and easy retrieval.

   The system must support CRUD (Create, Read, Update, Delete) operations efficiently through PHP backend scripts.

2. Non-Functional Requirements

Non-functional requirements describe **how the system performs** rather than what it does. These aspects ensure the system is efficient, secure, and maintainable.

1. **Performance and Responsiveness:**
   The system should load pages and process user requests within acceptable time limits.

   Proper use of optimized SQL queries and JavaScript will enhance speed and user experience.

2. **Security:**
   User authentication and session management will protect against unauthorized access.

   Passwords must be encrypted before storing in the database.

   Sensitive operations like payroll updates should only be accessible to authorized users.
3. **Usability:**
   The interface should be simple, intuitive, and visually appealing using **HTML, CSS, and JavaScript**.

   The navigation should be consistent across pages to ensure ease of use for all employees.
4. **Scalability:**
   The system should be designed to handle an increasing number of employees and records without major changes.

   The database structure should allow easy addition of new modules in the future.
5. **Compatibility:**
   WORKFUSION must be compatible with all major web browsers (Google Chrome, Firefox, Edge, Safari).

   The system should be responsive and accessible on both desktop and mobile devices.
6. **Maintainability:**
   The codebase should be modular and well-documented, making it easy for developers to maintain and update.

   Clear separation between front-end (HTML/CSS/JS) and back-end (PHP/MySQL) components should be maintained.
7. **Data Backup and Recovery:**
   Regular database backups should be maintained to prevent data loss in case of hardware or software failure.

   The system should support easy restoration of data from backup files.
8. **Reliability and Availability:**
   The system should remain functional with minimal downtime.
   Server errors or failures should be logged and handled gracefully to prevent data corruption.

9.SYSTEM DESIGN

SYSTEM DESIGN

The System Design phase is the blueprint of the entire Business Management System (WORKFUSION). It defines how the system will be structured, how data will flow between components, and how users will interact with it. This stage translates the previously gathered requirements into a clear technical model that serves as a guide for development and implementation.

System design focuses on two major aspects — high-level (architectural) design, which defines the overall structure, and detailed (component-level) design, which specifies the working of each module and database entity.

Objectives of System Design

The main objectives of designing the WORKFUSION system are:

- To establish a modular and scalable structure that supports future upgrades.
- To ensure efficient data flow between frontend, backend, and the database.
- To design a secure, user-friendly, and reliable interface for employees and administrators.
- To create a relational database model that maintains data integrity and minimizes redundancy.
- To ensure the system's architecture aligns with the chosen technology stack — HTML, CSS, JavaScript, PHP, and MySQL.

System Architecture

The overall architecture of WORKFUSION follows a Three-Tier Architecture Model, which divides the system into three main layers:

1. Presentation Layer (Frontend)

- Developed using HTML, CSS, and JavaScript.
- Responsible for user interaction and displaying system data in a clear, organized interface.
- Includes login forms, dashboards, employee records, attendance tables, leave forms, and reports.
- JavaScript is used for client-side validation and dynamic UI behaviour**.**

2. Application Layer (Business Logic)

- Implemented using PHP.
- Acts as the bridge between the frontend and the database.
- Handles core functionalities such as authentication, attendance calculation, payroll generation, leave validation, and report generation.

- Ensures that all user requests are processed securely and accurately before interacting with the database.

3. Data Layer (Database) • Powered by MySQL.

- Stores all system data including user credentials, employee details, attendance logs, leave requests, payroll data, and announcements.
- Maintains referential integrity using foreign keys and enforces constraints to prevent data inconsistency.
- This layered architecture improves security, modularity, and maintainability, ensuring that changes in one layer (e.g., frontend design) do not affect the others.

System Flow Design

The flow of operations in WORKFUSION is designed to be smooth and logical:

1. **User Authentication Flow**
   The user (employee or admin) logs in using their credentials.
   PHP verifies credentials from the MySQL database.
   If valid, the user is redirected to their respective dashboard; otherwise, an error message is shown.

2. **Employee Operation Flow**
   Employee can mark attendance, apply for leave, view payroll, and check announcements.
   Each action triggers a backend PHP script that updates or retrieves data from the database.

3. **Admin Operation Flow**
   The admin can manage employees, approve/reject leaves, schedule meetings, post announcements, and generate reports.
   All actions are logged in the database for future tracking and reporting.

4. **Report Generation Flow**
   The system retrieves data such as attendance and payroll from the database.
   PHP processes the data and displays reports on the admin dashboard, with options to export them in PDF or Excel formats.

Module Design

The WORKFUSION system is divided into several interconnected modules, each handling specific functions. This modular approach ensures maintainability and scalability.

1. Login and Authentication Module

- Handles secure login for employees and administrators.

- Uses PHP session management for user authentication.
- Prevents unauthorized access to sensitive pages.

2. Employee Management Module

- Allows admin to add, edit, delete, or view employee records.
- Stores employee information like name, designation, contact, and joining date.
- Ensures unique identification of employees using an Employee ID.

3. Attendance Management Module

- Enables employees to mark their daily attendance.
- Tracks login time and generates attendance statistics.
- Allows admin to view overall attendance and generate monthly reports.

4. Leave Management Module

- Employees can apply for leave online.
- Admin can approve or reject leave requests through the dashboard.
- The system maintains leave history and calculates available leave balance automatically.

5. Payroll Management Module

- Automatically calculates monthly salary based on attendance and leave deductions.
- Generates salary slips for each employee.
- Supports automatic rounding and tax/deduction calculations.

6. Meeting and Task Module

- Admin can schedule meetings and assign tasks to employees.
- Employees receive notifications about upcoming meetings.
- Helps improve internal coordination within the company.

7. Announcement and Notification Module

- Allows admin to broadcast announcements visible to all employees.
- Employees can view updates and notifications on their dashboard.

8.Reporting Module

- Generates analytical and statistical reports for admin.
- Includes attendance summary, leave utilization, and payroll breakdown.
- Supports exporting and printing reports for record-keeping.

Database Design

The database design forms the backbone of WORKFUSION. The MySQL relational model ensures data consistency and easy access across modules.

Key Tables:

1. Admin Table – stores admin login credentials.
2. Employee Table – holds employee personal and job-related information.
3. Attendance Table – records daily attendance status.
4. Leave Table – contains leave applications and approval status.
5. Payroll Table – stores salary details and deductions.
6. Meetings Table – records meeting details and schedules.
7. Announcements Table – holds messages and notifications posted by the admin.

Relationships:

Each employee record is linked to attendance, leave, and payroll tables using the Employee ID as a foreign key.

Admin actions like announcements and approvals are linked through the Admin ID. This relational structure minimizes redundancy and supports quick, efficient data retrieval.

User Interface (UI) Design

The UI is designed to be intuitive, responsive, and user-friendly. Using HTML5, CSS3, and JavaScript, the interface offers:

- A clean layout with clearly labeled menus and sections.
- Responsive design compatible with desktops, laptops, and tablets.32
- Consistent color schemes and typography for visual clarity.
- Interactive features like modals, validation alerts, and dynamic tables.
- Admins and employees have separate dashboards:
- Employee Dashboard: displays attendance status, leave balance, payroll info, and announcements.
- Admin Dashboard: provides control panels for managing employees, viewing reports, and system monitoring.

Security Design

Security is a core part of the system design. Key security measures include:

- Password encryption before storing in the database (using PHP's password_hash() function).
- Session-based authentication to prevent unauthorized access.
- Input validation and SQL injection prevention through prepared statements.
- Regular database backups to avoid data loss.

10.DATA FLOW DIAGRAMS

For the Business Management System (WORKFUSION), DFDs were prepared to visualize the structure of data flow and the logical relationships between the user, system, and processes.

Level 0 DFD

The Level 0 DFD, also known as the Context Diagram, provides a high-level overview of the entire system. It represents the system as a single process and shows how it interacts with external entities such as the Administrator and Employee.

Description:

The Administrator inputs data such as employee details, meeting information, announcements, and payroll rules into the system.
The Business Management System processes this information and provides outputs such as reports, salary details, and employee performance data.
The Employee interacts with the system by marking attendance, applying for leave, and viewing announcements or salary slips.
The system maintains all data in the database and generates responses to both the administrator and employee based on their roles.

Key Data Flows:

From Administrator → System: Employee data, meeting details, notices, and approvals.
From Employee → System: Attendance records, leave requests, and login details.
From System → Administrator: Attendance reports, payroll summaries, and employee performance details.
From System → Employee: Leave status, salary slips, and announcements

Level 1 DFD (Detailed System Flow)

The Level 1 DFD expands the single process of the Level 0 DFD into multiple sub-processes, showing how data flows internally between the modules of the system.
In WORKFUSION, the main process is divided into several smaller processes:

1. Employee Management

- Handles employee registration, updates, and personal information.
- Admin enters employee data, which is stored in the database and used by other modules.

2. Attendance Tracking

- Manages daily check-in and check-out data of employees.
- Send attendance records to the payroll module for salary calculation and to the admin for reports.

3. Leave Management

- Allow employees to apply for leave and tracks available leave limits.
- Sends approved leave data to the payroll module for deductions.

4. Payroll and Deductions

- Generates salary based on attendance and leave data.
- Calculates deductions automatically and stores the payroll details in the database.

5. Meeting and Announcement Management

- Enables the admin to schedule meetings and post announcements.
- Displays updates to all employees through their dashboards.

Data Stores (Databases):

Employee Database: Stores all employee details.

Attendance Database: Records daily attendance.

Leave Database: Maintains leave applications and approvals.

Payroll Database: Contains salary and deduction data.

Announcement Database: Keeps meeting and notice details.

Data Flow Summary:

The Employee provides data such as attendance and leave requests.

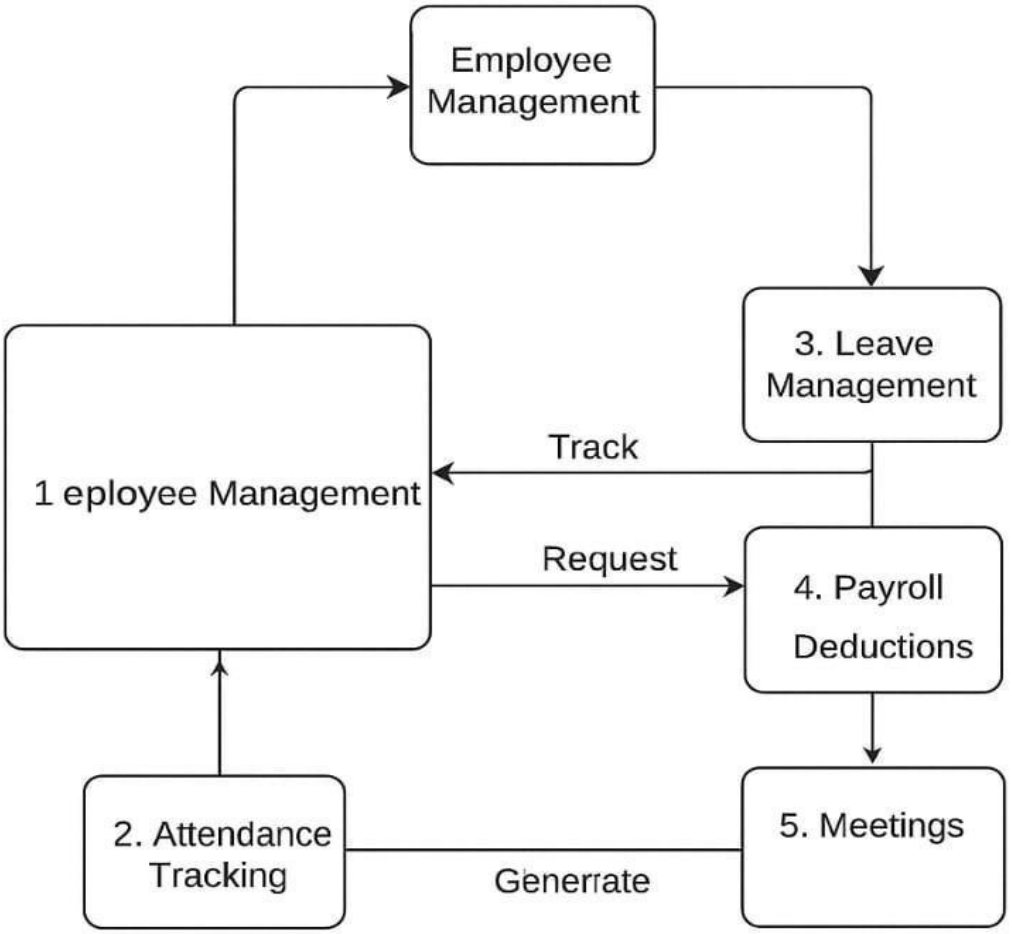
The Administrator inputs and manages organizational data.

Each module exchanges data internally to perform tasks like payroll calculation and reporting.

Final results are displayed back to both the administrator and employees in the form of dashboards and reports.

Level 0

Level 1



11.Database Design

The system uses several relational tables interconnected via primary and foreign keys. Major tables include:

• users (user_id, name, role, email, password)

• attendance (id, user_id, date, checkin_time, checkout_time)

• leaves (id, user_id, leave_type, from_date, to_date, status)

• payroll (id, user_id, month, basic, deduction, net_salary)

• meetings (id, title, date, time, attendees)

• tasks (id, task_name, assigned_to, status)

• announcements (id, title, description, posted_date)

12.Implementation

The implementation phase of the Business Management System (WORKFUSION) is the most crucial stage of the software development process. It involves transforming the system design and database structure into a fully functional application. In this phase, all modules and components were coded, integrated, tested, and deployed in a real-time environment to ensure that the system performs as expected.

The implementation was carried out after careful requirement analysis, system design, and testing of individual modules. The goal was to create a secure, efficient, and user-friendly management platform that automates business operations like employee management, attendance tracking, leave handling, payroll generation, meeting scheduling, and announcement management.

1. Implementation Environment

The system was developed in a local server environment using the XAMPP package, which includes Apache Server, PHP, and MySQL. The frontend was created using HTML, CSS, and JavaScript, while the backend logic was implemented using PHP.

All coding and interface design were done using Visual Studio Code (VS Code) a powerful development environment with debugging and syntax highlighting features. MySQL database management was done through phpMyAdmin, which comes integrated with XAMPP. The project was tested locally on a Windows operating system, using Google Chrome as the primary browser. After successful local testing, the system was made ready for deployment on a live web server for online access.

2. Module-Wise Implementation

The WORKFUSION system consists of several interconnected modules. Each module was implemented step-by-step and tested individually before integration to ensure smooth operation.

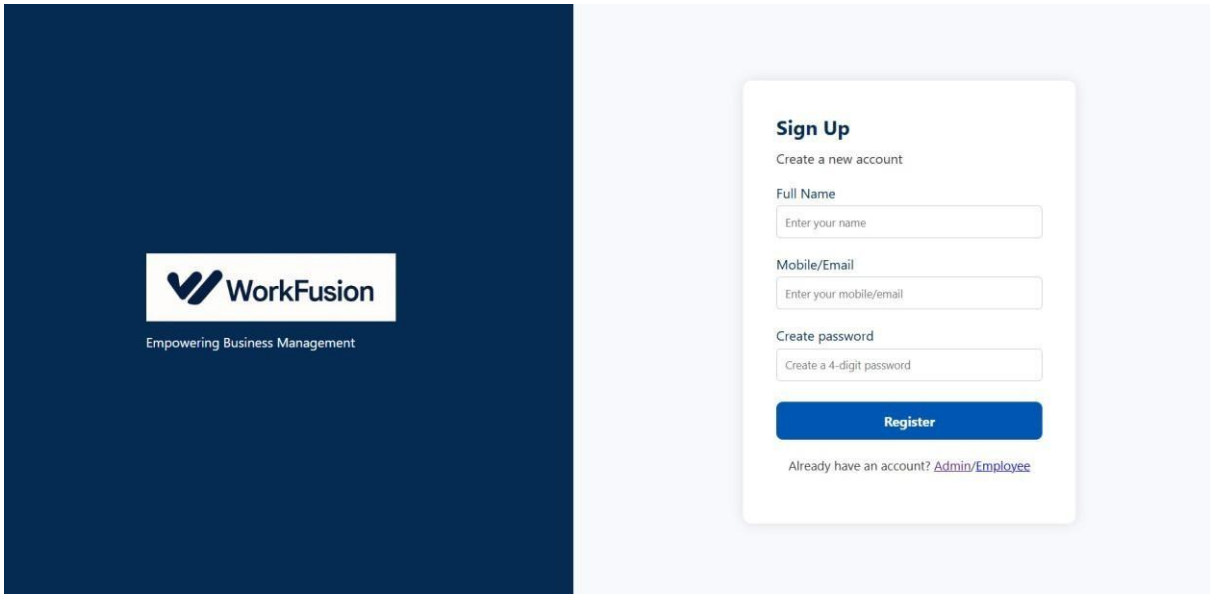
a) Login and Authentication Module

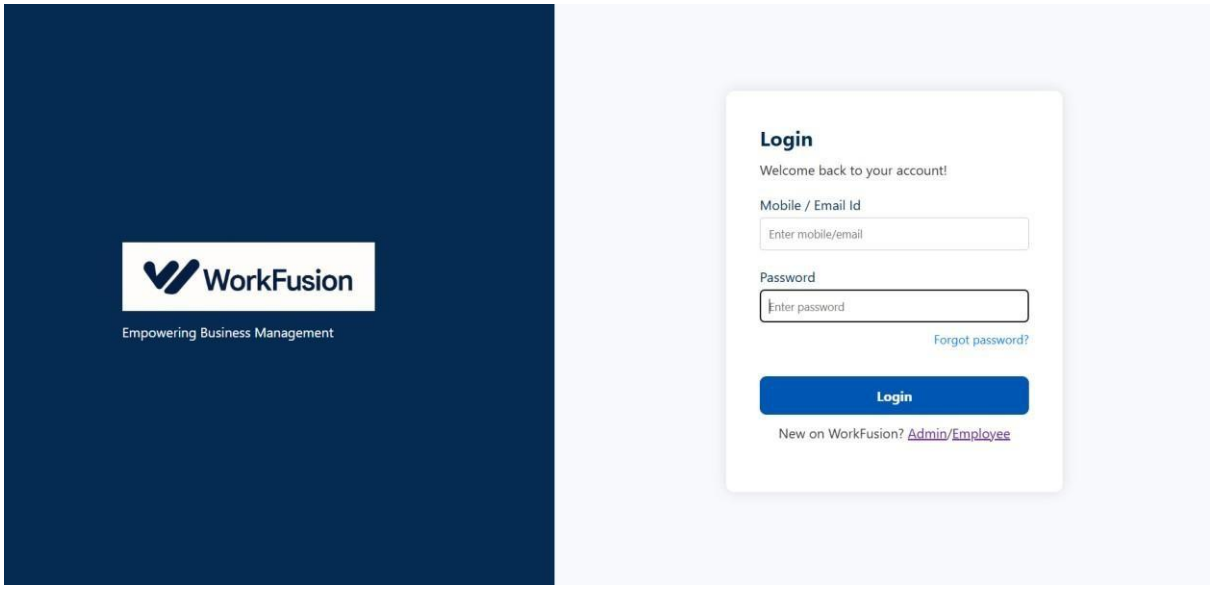
This is the entry point of the system where both administrators and employees log in using their credentials.

Admins have access to all management functions.

Employees have limited access to their dashboard and personal records.

User authentication ensures data privacy and prevents unauthorized access.

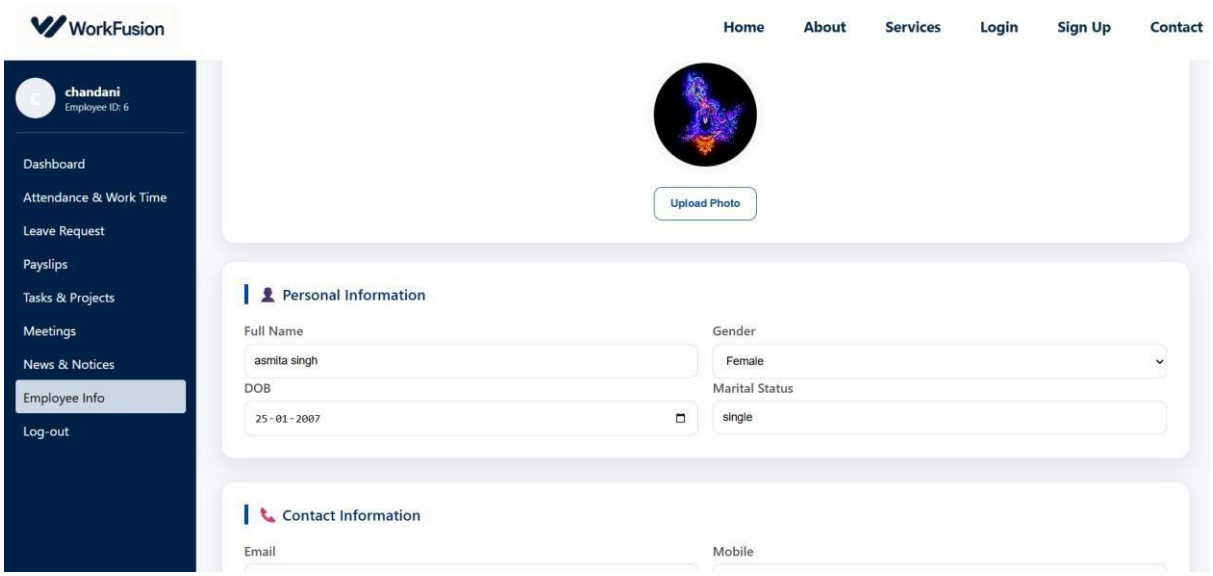

WorkFusion
Empowering Business Management
Sign Up
Create a new account
Full Name
Enter your name
Mobile/Email
Enter your mobile/email
Create password
Create a 4-digit password
Register
Already have an account? Admin/Employee

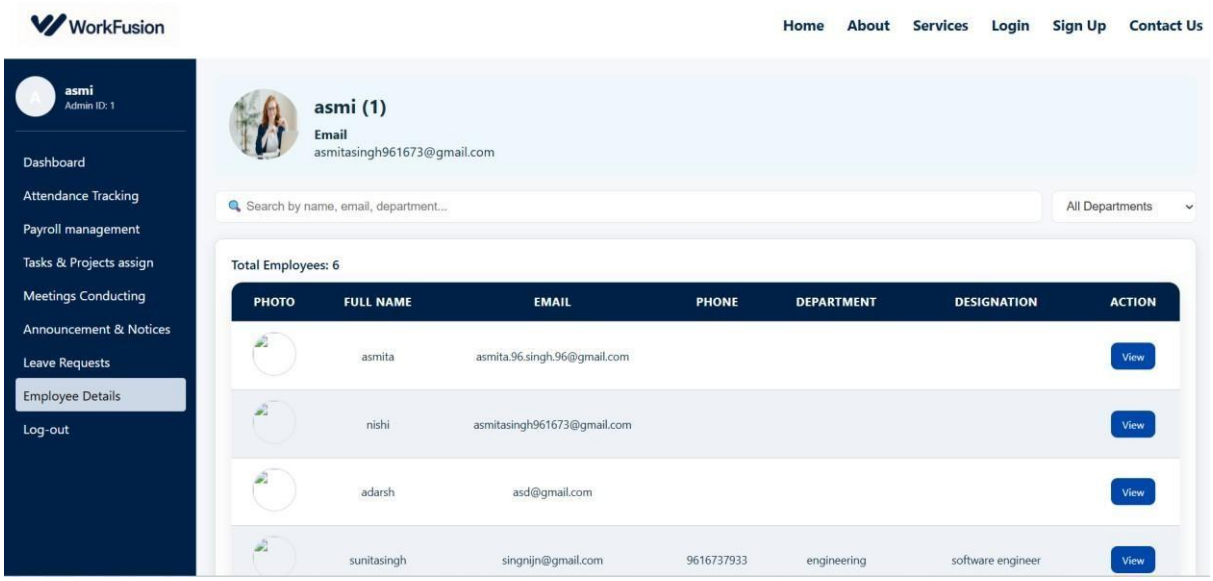

WorkFusion
Empowering Business Management
Login
Welcome back to your account!
Mobile / Email Id
Enter mobile/email
Password
Enter password
Forgot password?
Login
New on WorkFusion? Admin/Employee

b) Employee Management Module

This module allows administrators to add, edit, or delete employee details such as name, department, designation, salary, and contact information. Employees can view and update limited information in their profile.

This module acts as the foundation for other components like attendance, leave, and payroll.

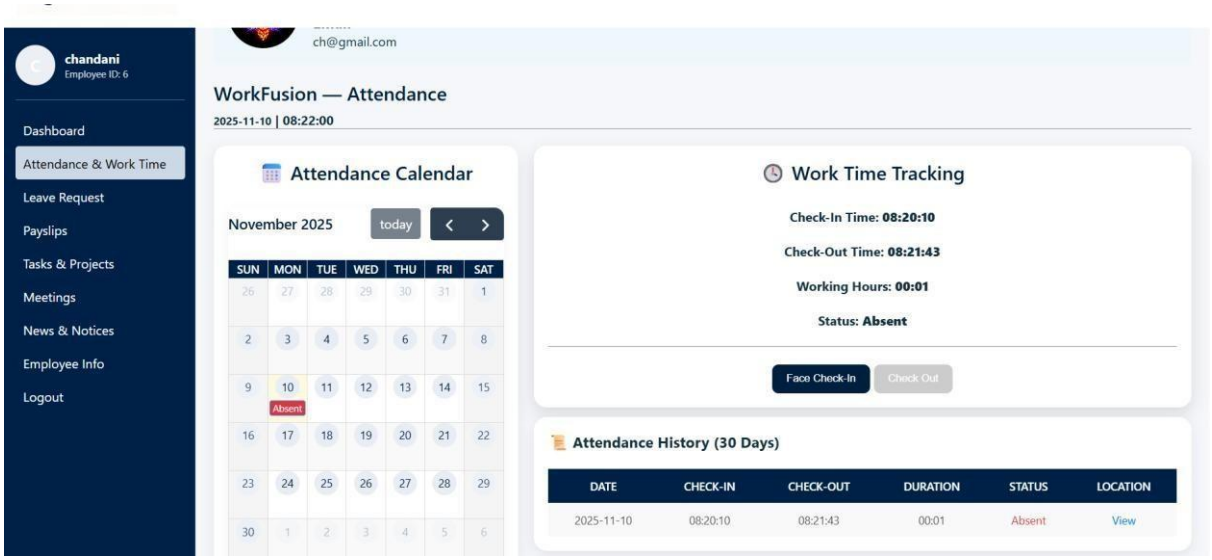


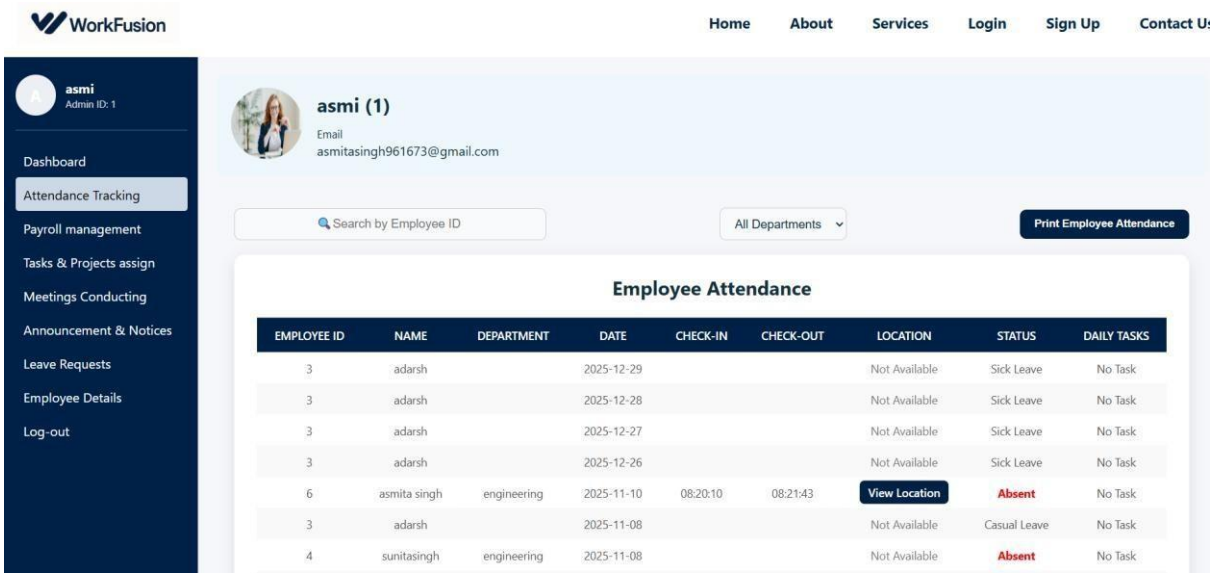


c) Attendance Tracking Module

The attendance system was implemented to allow employees to check in and check out digitally. If an employee fails to mark attendance before a certain time (e.g., 11:00 AM), the system automatically marks them as being absent.

This module uses PHP scripts to capture time data and update the attendance table in the database. The admin can view daily and monthly attendance reports of all employees.

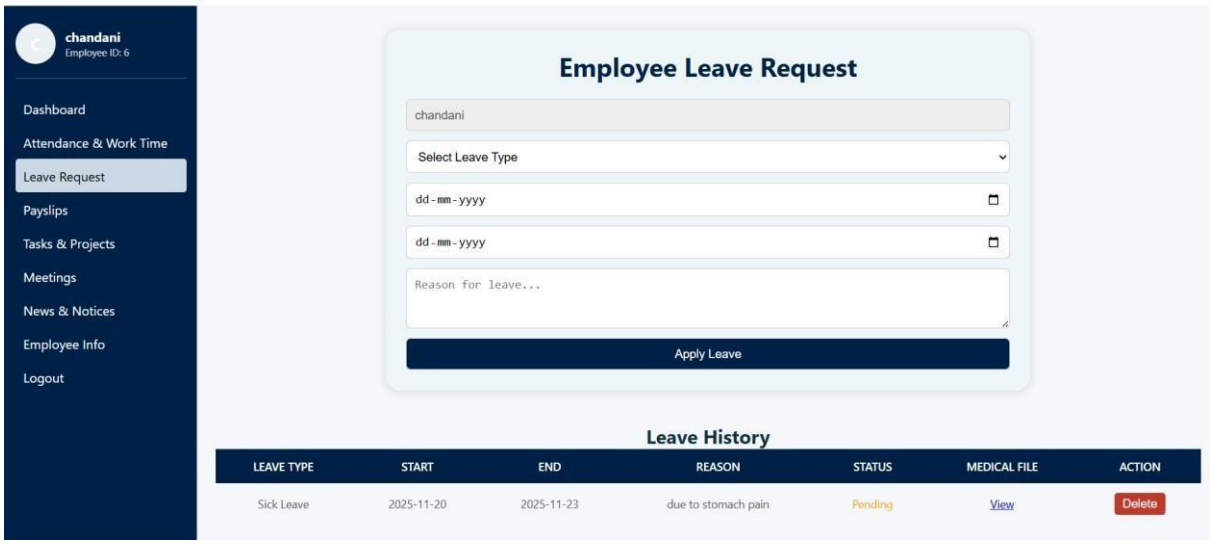


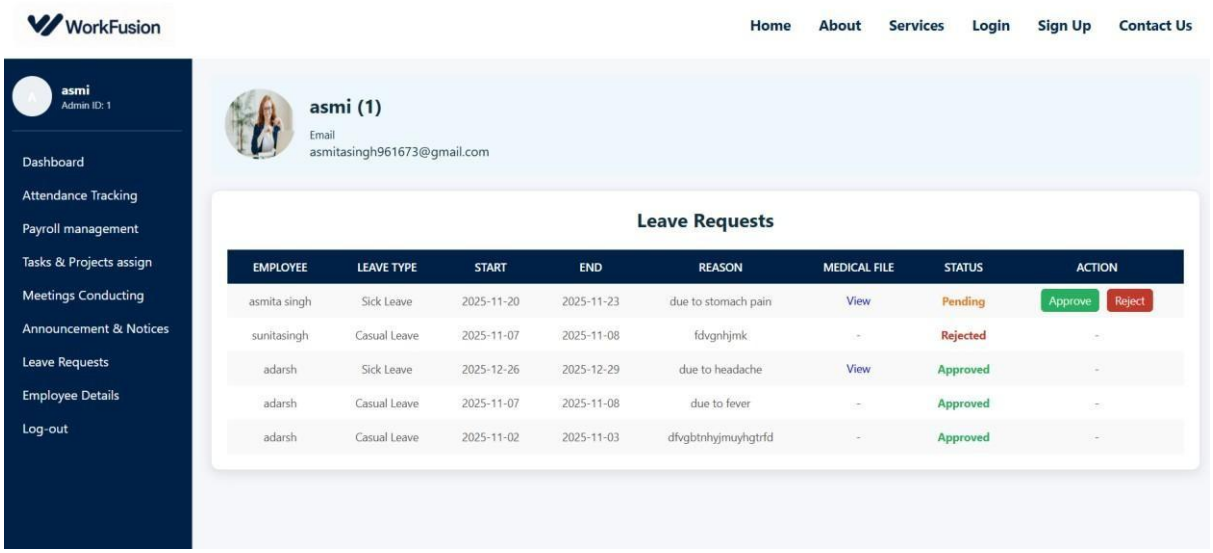


d) Leave Management Module

The leave management system allows employees to apply for different types of leaves (Casual, Sick, Paid). Leave limits are predefined, and the system generates alerts when the employee exceeds the allowed limit.

Admins can approve or reject leave requests directly from their panel. This module connects with the attendance and payroll systems to ensure accurate salary deductions.

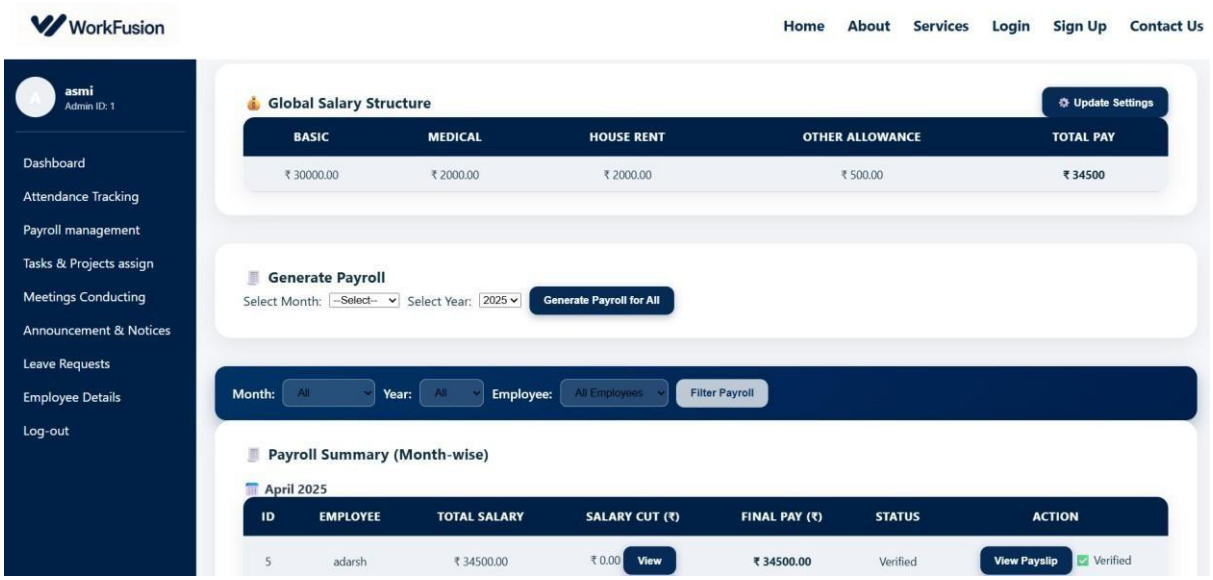


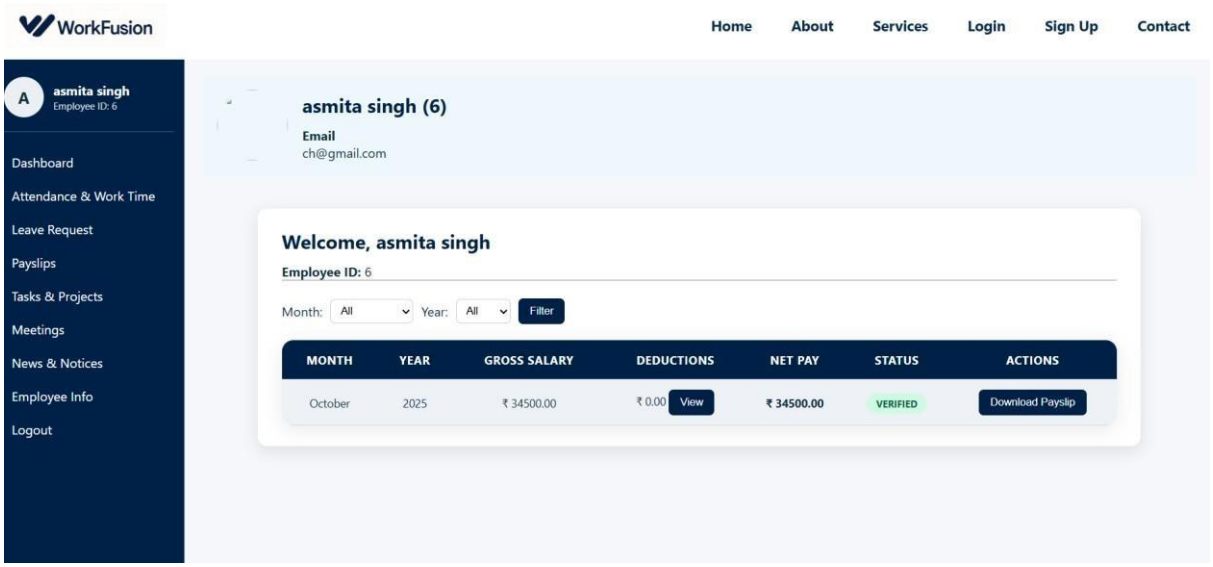


e) Payroll and Deduction Module

This module automatically calculates each employee's salary based on attendance, leave data, and deductions. It considers allowances and generates a detailed monthly summary that both employees and admins can view.

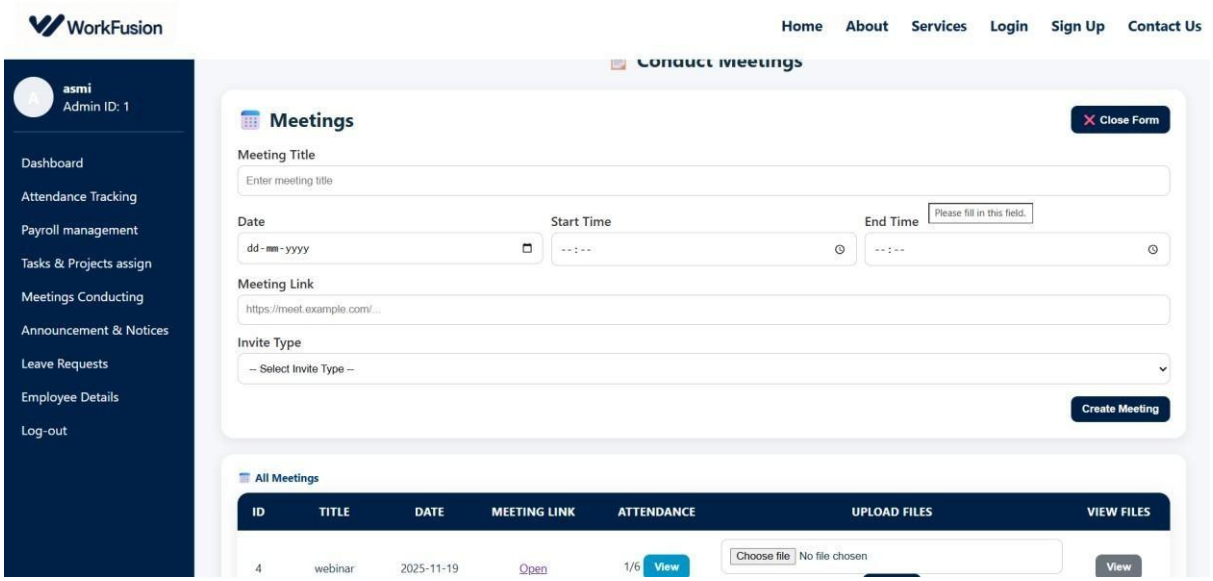
The system also includes a Deduction Summary section that records fines, late arrivals, or excess leaves. The payroll module ensures complete transparency and accuracy in salary calculations.

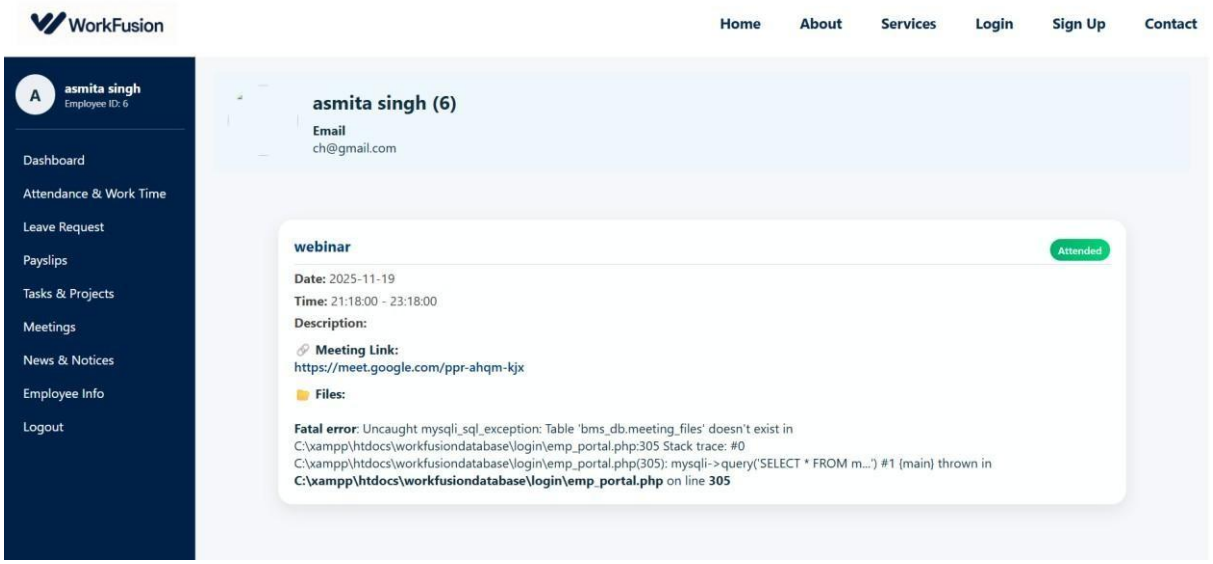




f) Meeting Scheduling and Task Assignment Module

In this module, the administrator can create meeting schedules, assign specific projects or tasks to employees, and set deadlines.
Employees can view their upcoming meetings and assigned projects in their dashboard. The module helps in maintaining organizational discipline and coordination.

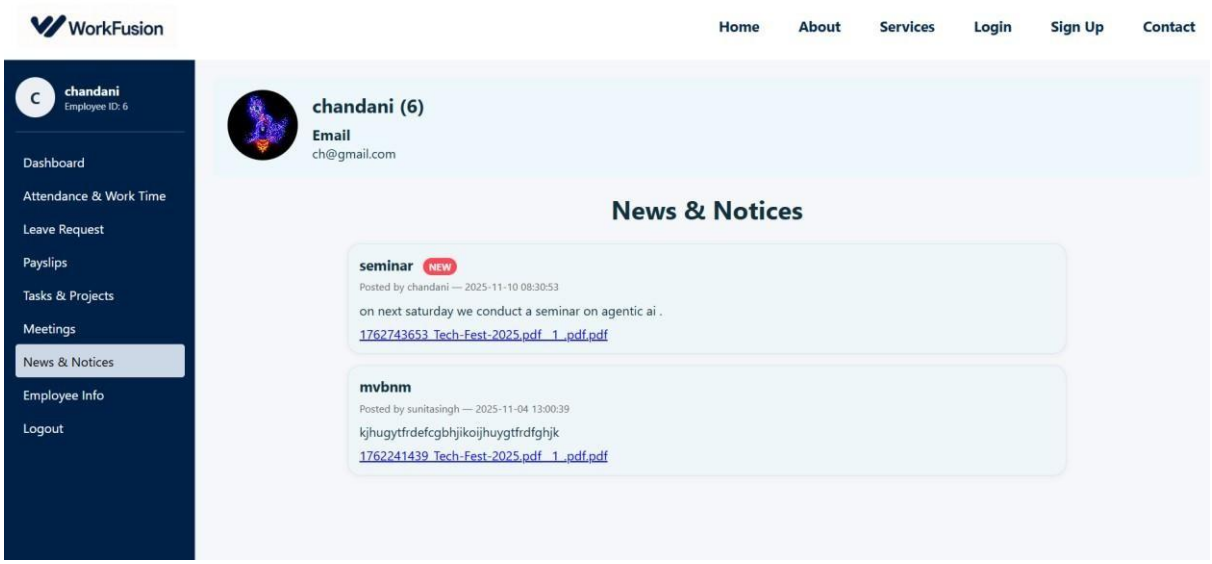




g) Announcements and Notices Module

This module allows the admin to post official notices, upcoming events, or important company news. All announcements are visible on the employee dashboard.

This ensures quick internal communication and reduces dependency on manual notifications.



3. Integration and System Testing

After all individual modules were implemented successfully, they were integrated into a unified system. Data exchange between modules — such as attendance affecting payroll or leave records updating salary calculations — was thoroughly tested.

The testing phase ensured that every module worked correctly both individually and collectively. User feedback from initial testing was used to make interface improvements and optimize performance.

4. User Interface Implementation

A lot of focus was placed on designing a simple, modern, and responsive interface. The use of CSS and JavaScript made the dashboards interactive and visually appealing. All key information — such as employee details, salary summaries, leave balance, and meeting alerts — is easily accessible from the main dashboard.

The interface was tested on different screen sizes and browsers to ensure compatibility and smooth navigation.

13. Code

In this section, the complete source code of the Business Management System (BMS) is presented. The purpose of including the code is to provide a clear understanding of the system's internal working, logic implementation, and structure. The code demonstrates how different modules of the application interact with each other to perform key business operations such as employee management, attendance handling, leave tracking, project monitoring, and report generation.

The source code is divided into multiple files based on functionality, which ensures modularity, reusability, and easy maintenance. Each code segment has been written using standard programming practices such as proper indentation, meaningful variable names, comments, and secure database operations.

1. Signup

```
<?php include
'server.php';
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<title>Sign Up - BMS</title>
<link rel="stylesheet" href="signup.css" />
</head>
<body>
<div class="container">
  <div class="left-section">
    <div class="logo">
      <img src="logo.png" alt="Logo" class="logo-img" />
      <p>Empowering Business Management</p>
    </div>
  </div>
  <div class="right-section">
    <div class="signup-box">
      <h2>Sign Up</h2>
      <p>Create a new account</p>
      <form action="signup.php" method="POST">
        <?php include 'error.php'; ?>

        <label for="name">Full Name</label>
        <input type="text" name="name" placeholder="Enter your name" required />
        <label for="mobile">Mobile/Email</label>
```

```
            <input type="text" name="mobile" placeholder="Enter your mobile/email" required
/>
            <label for="mpin">Create password</label>
            <input type="password" name="mpin" placeholder="Create a 4-digit password"
required />
            <button type="submit" class="signup-btn" name="reg_user">Register</button>
            <p class="login-link">Already have an account? <a
href="admin_login.php">Admin</a><span>/</span><a
href="login.php">Employee</a></p>
          </form>
       </div>
    </div>
</div>
<script src="script.js"></script>
</body>
</html>
```

2. Login

```
<?php          include     "Connect.php";
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST['email'];
    $password = $_POST['password'];
    $sql = "SELECT * FROM admins WHERE mobile_email='$email'";
    $result = $conn->query($sql);     if
($result->num_rows == 1) {
$admin   =   $result->fetch_assoc();                    if   (password_verify($password,
$admin['password'])) {            $_SESSION['admin_id'] = $admin['id'];
          $_SESSION['admin_name']                =                $admin['full_name'];
$_SESSION['admin_email']              =              $admin['mobile_email'];
header("Location: admin.php");
exit;         } else {
          echo "<script>alert('Invalid Password!');</script>";
       }
    }  else  {              echo  "<script>alert('Admin  not
found!');</script>";
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>  <title>Login
- BMS</title>
```

```
<link rel="stylesheet" href="style.css" />
</head>
<body>
<div class="container">
   <div class="left-section">
      <div class="logo">
         <img src="logo.png" alt="Logo" class="logo-img" />
         <p>Empowering Business Management</p>
      </div>
   </div>
   <div class="right-section">
      <div class="login-box">
         <h2>Admin Login</h2>
         <p>Welcome back to your account!</p>
         <form action="admin_login.php" method="POST">
            <?php include('error.php'); ?>
            <label for="userId">Mobile / Email Id  </label>
            <input type="text" name="email" placeholder="Enter mobile/email" required />
            <label for="password">Password</label>
            <input type="password" name="password" placeholder="Enter password" required />
            <a href="forgot_pass.php" class="forgot-link">Forgot password?</a>
            <button type="submit" class="login-btn"  name="login_user">Login</button>
            <p class="register">New on WorkFusion? <a
href="admin_signup.php">Admin</a><span>/</span><a
href="signup.php">Employee</a></p>
         </form>
         </form>
      </div>
   </div>
</div>
<script src="script.js"></script>
</body>
</html>
```

3.Navbar

```
 <header>
<nav class="navbar">
  <img src="logo.png" alt="WorkFusion Logo" class="logo" />
  <div class="nav-links" id="navLinks">
    <a href="home1.html">Home</a>
    <a href="about.html">About</a>
    <a href="services.html">Services</a>
    <a href="login.php">Login</a>
    <a href="signup.php">Sign Up</a>
    <a href="contact.html">Contact</a>
```

```
  </div>
  <div class="hamburger" id="hamburger">
    <span></span><span></span><span></span>
  </div>
 </nav>
</header>
```

4. Employee Sidebar Menu

```
<div class="app">
  <aside class="sidebar">
    <div class="user">
      <div class="avatar">
        <?php echo isset($_SESSION['full_name']) ?
strtoupper(substr($_SESSION['full_name'],0,1)) : '?'; ?>
      </div>
      <div class="user-info">
        <div class="name"><?php echo $_SESSION['full_name'] ?? 'Guest'; ?></div>
        <div class="emp-id">Employee ID: <?php echo $_SESSION['emp_id'] ?? 'N/A';
?></div>
      </div>
    </div>


    <hr class="sidebar-separator">
    <?php $current_page = basename($_SERVER['PHP_SELF']); ?>
    <nav class="menu">

      <a href="employee.php" class="<?= $current_page=='employee.php'?'active':"
?>">Dashboard</a>
      <a href="employee_worktimetracking.php" class="<?=
$current_page=='employee_worktimetracking.php'?'active':" ?>">Attendance & Work
Time</a>
      <a href="employee_leave.php" class="<?=
$current_page=='employee_leave.php'?'active':" ?>">Leave Request</a>
      <a href="employee_payroll.php" class="<?=
$current_page=='employee_payroll.php'?'active':" ?>">Payslips</a>
      <a href="employee_tasks_projects.php" class="<?=
$current_page=='employee_tasks_projects.php'?'active':" ?>">Tasks & Projects</a>
      <a href="emp_portal.php" class="<?= $current_page=='emp_portal.php'?'active':"
?>">Meetings</a>
      <a href="employee_news.php" class="<?=
$current_page=='employee_news.php'?'active':" ?>">News & Notices</a>
      <a href="employee_profile.php" class="<?=
$current_page=='employee_profile.php'?'active':" ?>">Employee Info</a>
      <a href="logout.php">Logout</a>
    </nav>
  </aside>
```

5. Admin Sidebar Menu

```
   <nav class="menu">
     <a href="admin.php" class="<?= $current_page=='admin.php' ? 'active' : "
?>">Dashboard</a>
       <a href="admin_att_tracking.php"  class="<?=
$current_page=='admin_att_tracking.php'  ?  'active'  :  "  ?>">Attendance  Tracking</a>
<a href="admin_payroll.php"   class="<?= $current_page=='admin_payroll.php' ?
'active' : " ?>">Payroll management</a>
       <a href="admin_projects.php"  class="<?= $current_page=='admin_projects.php' ?
'active' : " ?>">Tasks & Projects assign</a>
       <a href="admin_portal.php"  class="<?= $current_page=='admin_portal.php' ?
'active' : " ?>">Meetings Conducting</a>
       <a href="admin_notices.php"  class="<?= $current_page=='admin_notices.php' ?
'active' : " ?>">Announcement & Notices</a>
       <a href="admin_leave.php"  class="<?= $current_page=='admin_leaves.php' ?
'active' : " ?>">Leave Requests</a>
       <a href="admin_emp.php"  class="<?= $current_page=='admin_emp.php' ? 'active' :
" ?>">Employee Details</a>



       <a href="logout.php"  class="<?= $current_page=='logout.php' ? 'active' : "
?>">Log-out</a>
     </nav>
```

6. Employee Attendance Management System

```
 <div class="layout">
   <div>
     <div class="card">
       <center><h2>  Attendance Calendar</h2></center>
       <br>
       <div id="calendar"></div>

       <!-- Color Definition Legend -->
       <div class="legend">
         <div class="legend-item"><span class="legend-color"
style="background:#28a745;"></span>          Present</div>
<div     class="legend-item"><span     class="legend-color"
style="background:#ffc107;"></span>    Half    Day</div>
<div     class="legend-item"><span     class="legend-color"
style="background:#dc3545;"></span>          Absent</div>
<div     class="legend-item"><span     class="legend-color"
```

```
style="background:#6f42c1;"></span>        Overtime</div>
<div      class="legend-item"><span      class="legend-color"
style="background:#0dcaf0;"></span> Leave</div>
       </div>
      </div>
      <div class="card" style="margin-top:12px">
       <h3>  Last Check-In Location</h3>
       <div id="mapContainer"><small>No recent check-in</small></div>
      </div>
    </div>
    <div>
      <div class="card">
       <center><h2>  Work Time Tracking</h2></center>
       <br>
       <center><h4><p>Check-In                Time:                <strong                id="ci">--:-
</strong></h4></p></center>
        <br>
       <center><h4><p>Check-Out               Time:                <strong               id="co">--:--
</strong></p></h4></center>
        <br>
       <center><h4><p>Working Hours: <strong id="wd">--
</strong></p></h4></center>
        <br>
       <center><h4><p>Status: <strong id="st">--</strong></p></h4></center>
        <br>
       <hr>
        <br>
       <center><button id="checkinBtn" class="btn">Face Check-In</button>
<button id="checkoutBtn" class="btn" disabled>Check Out</button></center>   </div>
      <div class="card" style="margin-top:12px">
       <h3>  Attendance History (30 Days)</h3>
       <br>
       <div id="historyBox">Loading...</div>
      </div>
    </div>
  </div>

  <!--   Face Modal -->
<div id="faceModal">
  <div class="modal-card">
    <h3 style="color:var(--primary);font-weight:600;">  Face Check-In  Verification</h3>
    <video id="video" autoplay playsinline></video>
    <div style="margin-top:12px;">
      <button id="captureBtn" class="btn">  Capture & Check-In</button>
      <button   id="closeModal"   class="btn"   style="background:#555;">Cancel</button>
</div>
    <p style="font-size:13px;color:#666;margin-top:8px;">Check-in before <b>11:00
AM</b> is required.</p>
  </div>
```

```
  </main>
</div>

<script>  document.addEventListener('DOMContentLoaded',
function() {   const checkinBtn =
document.getElementById('checkinBtn');   const checkoutBtn =
document.getElementById('checkoutBtn');   const faceModal =
document.getElementById('faceModal');   const video =
document.getElementById('video');   const captureBtn =
document.getElementById('captureBtn');   const closeModal =
document.getElementById('closeModal');


  const ciEl = document.getElementById('ci'), coEl = document.getElementById('co'),
wdEl = document.getElementById('wd'), stEl = document.getElementById('st');   const
historyBox = document.getElementById('historyBox');

  function      updateClock()      {
const now = new Date();
    document.getElementById('currentDate').textContent          =
now.toLocaleDateString('enCA');
    document.getElementById('currentTime').textContent = now.toLocaleTimeString();
const day = now.getDay();

    if (day === 0 || day === 6) {   checkinBtn.disabled
    = true;      checkinBtn.textContent = "Weekend — No
    Check-In";
    }                 else                 {
checkinBtn.disabled                    =                   false;
checkinBtn.textContent = "Face Check-In";
    }
  }
  setInterval(updateClock,          1000);   updateClock();

  const calendar = new FullCalendar.Calendar(document.getElementById('calendar'), {
initialView: 'dayGridMonth',
    height: 450,
    dayCellDidMount: function(info) {
const day = info.date.getDay();       if (day
=== 0 || day === 6) {
       info.el.style.backgroundColor                =                '#f3f3f3';
info.el.style.opacity = '0.8';
      }
    }   });
calendar.render();

  async function loadData() {     const res = await
fetch('attendance_backend.php?action=fetch');     const j = await res.json();
    if (j.status !== 'success') return;
```

```
    const events = [];
  j.data.forEach(r => {   const
dateObj = new Date(r.date);
  const day = dateObj.getDay(); // 0 = Sunday, 6 = Saturday

  //   Skip showing leave or attendance entries on weekends   if
(day === 0 || day === 6) return;

  let color = '#adb5bd';   if (r.status === 'Present')
color = '#28a745';   else if (r.status === 'Half Day')
color = '#ffc107';   else if (r.status ===
'Absent') color = '#dc3545';   else if (r.status ===
'Overtime') color = '#6f42c1';   else if (r.status ===
'Leave') color = '#0dcaf0';

  events.push({
    title: r.status,

    start: r.date, allDay:
    true,
    backgroundColor: color
  });
});

    calendar.removeAllEvents();
calendar.addEventSource(events);

   let html = `
  <table style="width:100%;border-collapse:collapse;">
    <tr>
      <th>Date</th>
      <th>Check-In</th>
      <th>Check-Out</th>
      <th>Duration</th>
<th>Status</th>
      <th>Location</th>
    </tr>
`;
 j.data.forEach(r     =>     {
html += `<tr>
  <td>${r.date}</td>
  <td>${r.checkin_time || '--'}</td>
  <td>${r.checkout_time || '--'}</td>
  <td>${r.work_duration || '--'}</td>
  <td>
    <span style="color:${
      r.status === 'Present' ? '#28a745' :
      r.status === 'Half Day' ? '#ffc107' :
      r.status === 'Absent' ? '#dc3545' :
```

```
      r.status === 'Overtime' ? '#6f42c1' : '#0dcaf0'
    }">${r.status || '--'}</span>
  </td>
  <td>
  ${
    r.latitude && r.longitude
      ?       `<a       href="https://www.google.com/maps?q=${r.latitude},${r.longitude}"
target="_blank" style="color:#007bff;text-decoration:none;">View</a>`
      : '--'
  }
</td>
</tr>`;
    });          html  +=
'</table>';
    historyBox.innerHTML = html;


  } loadData(); let stream
  = null;
  checkinBtn.addEventListener('click', async () => {
    if (checkinBtn.disabled) return;
try {
      stream = await navigator.mediaDevices.getUserMedia({ video: true });
video.srcObject = stream;       faceModal.style.display = 'flex';
    } catch {            alert('Camera
access denied');
    }
  });

  closeModal.addEventListener('click', () => {
faceModal.style.display = 'none';     if (stream)
stream.getTracks().forEach(t => t.stop());
  });
  captureBtn.addEventListener('click', async () => {     const
canvas = document.createElement('canvas');     canvas.width
= video.videoWidth;     canvas.height = video.videoHeight;
canvas.getContext('2d').drawImage(video, 0, 0);     const
imageData = canvas.toDataURL('image/png');
navigator.geolocation.getCurrentPosition(async (pos) => {
const fd = new FormData();       fd.append('action',
'face_checkin');       fd.append('image', imageData);
fd.append('latitude', pos.coords.latitude);
fd.append('longitude', pos.coords.longitude);
      const resp = await fetch('attendance_backend.php', { method: 'POST', body: fd });
const j = await resp.json();       alert(j.message);       if (j.status === 'success') {
ciEl.textContent = j.checkin_time;         checkoutBtn.disabled = false;
faceModal.style.display = 'none';
        if (stream) stream.getTracks().forEach(t => t.stop());
        //   Show map          const { latitude, longitude } =
pos.coords;
```

```
      const mapContainer = document.getElementById('mapContainer');
mapContainer.innerHTML = `
        <iframe width="100%" height="200" style="border:0; border-radius:10px;"
loading="lazy" allowfullscreen
src="https://www.google.com/maps?q=${latitude},${longitude}&hl=en&z=15&output
=embed"></iframe>
        <p style="font-size:13px;color:#555;margin-top:4px;">
          Location captured at ${latitude.toFixed(4)}, ${longitude.toFixed(4)}        </p>`;
    }
   });
  });
```

7. Employee Payslip

```
<?php  session_start();
include "Connect.php";

$emp_id = $_SESSION['emp_id'];
$user = $conn->query("SELECT * FROM users WHERE id=$emp_id")>fetch_assoc();

$emp_name = $user['full_name'];
$basic_salary = $user['basic_salary'] ?? 0;

// ======= Fetch Payroll Records ======= // $where
= "WHERE p.emp_id=$emp_id";  if (!empty($_GET['year_filter'])) $where .= " AND
p.year='{$_GET['year_filter']}'"; if (!empty($_GET['month_filter'])) $where .= " AND
p.month='{$_GET['month_filter']}'";

$payrolls = $conn->query("
   SELECT p.*, u.full_name
   FROM payroll p
   JOIN users u ON p.emp_id=u.id
   $where
   ORDER BY p.year DESC,
   FIELD(p.month,
'January','February','March','April','May','June','July','August','September','October','Nov
ember','December')
");
```

14.Testing

Software testing is an essential phase in the Software Development Life Cycle (SDLC) that ensures the system performs according to user requirements and functions correctly under all conditions. The testing process for the Business Management System (WORKFUSION) was carried out systematically to detect and correct errors, verify the accuracy of each module, and confirm that the system meets its objectives.

The primary goal of testing was to ensure that the application is reliable, secure, userfriendly, and performs efficiently when handling multiple user operations such as attendance marking, leave management, payroll processing, and meeting scheduling.

• Types of Testing Performed

Different levels and types of testing were carried out to ensure the system's correctness and stability. The testing process included Unit Testing, Integration Testing, System Testing, User Acceptance Testing (UAT), and Performance Testing.

a) Unit Testing

Each module of the system was tested individually to verify that it performs its intended functions correctly. For example:
The login module was tested to ensure valid credentials grant access while invalid inputs are rejected.
The attendance module was checked to confirm that data is recorded accurately with correct timestamps.
The leave management module was tested to ensure that leave requests follow proper validation and limit rules.
Unit testing helped identify early-stage errors in scripts and logic, which were then corrected before integration.

b) Integration Testing

After successful unit testing, integration testing was conducted to ensure smooth communication between modules. For instance, leave records and attendance data were tested to confirm that they update the payroll module correctly. Similarly, announcements and meeting data were checked for proper display in the employee dashboard. Integration testing validated that modules worked together without conflicts.

c) System Testing

System testing was carried out after integrating all modules to check the complete functionality of the application. The entire WORKFUSION system — including login, attendance, leave, payroll, meetings, and announcements — was tested in real-time scenarios. This testing ensured that the application performed well as a whole and that all workflows functioned smoothly from start to end.

d) User Acceptance Testing (UAT)

In this phase, the system was tested by actual users — both administrators and employees — to verify that it meets their practical needs. User feedback was collected on system performance, ease of use, and design clarity. Based on this feedback, some interface improvements and alert messages were added for better user experience. The users confirmed that the system simplified their daily operations and improved management efficiency.

e) Performance Testing

Performance testing focused on evaluating the system's speed, response time, and stability when handling multiple users or large amounts of data. The database was tested for quick data retrieval, and the web interface was monitored for page loading time. The results showed that the system could handle concurrent users efficiently without significant delay.

f) Security Testing

Security testing was carried out to ensure data protection and restricted access. Login validation, password encryption, and role-based permissions were tested thoroughly. Attempts to access admin pages from employee accounts were successfully blocked, ensuring that sensitive data remains secure.

15. Future Scope

The **Business Management System (WORKFUSION)** has been built with a modular, scalable, and secure architecture to support long-term adaptability. While the current version successfully automates key business operations like employee management, attendance, payroll, and leave tracking, there remains significant potential for further development. This section outlines the **future enhancements** that can make the system more intelligent, userfriendly, and enterprise ready. The integration of modern technologies like cloud computing, artificial intelligence, and biometrics can transform WORKFUSION into a comprehensive and fully automated business management solution.

1. Biometric Attendance Integration:

One of the most practical and secure future upgrades for WORKFUSION is **biometric attendance tracking**.

By integrating **fingerprint scanners or facial recognition technology**, the system can automatically record employee attendance without manual input.

This will help:

- Eliminate **proxy or false attendance entries**.
- Increase the **accuracy of working hours** and overtime calculations.
- Improve **overall accountability and transparency** in attendance management. Biometric data can also be linked directly with the payroll and leave management modules, ensuring that attendance discrepancies are minimized.

2. Cloud Hosting:

Currently, the system may operate on a local server within the organization. In the future, **deploying WORKFUSION on cloud platforms** like **Amazon Web Services (AWS), Microsoft Azure, or Google Cloud** will enhance flexibility and reliability.

Cloud hosting will provide:

- **Anywhere, anytime access** for administrators and employees.
- **Automatic data backups** and disaster recovery options.
- **Scalability** to handle an increasing number of users or organizations.
- **Reduced maintenance costs** compared to on-premise hosting.

This transition will make the system suitable for medium to large enterprises operating from multiple locations.

3. Mobile Application:

In the modern workplace, accessibility on-the-go is essential. The development of **mobile applications for Android and iOS** will allow users to manage their daily tasks seamlessly.

Employees can use the mobile app to:

- Mark attendance remotely (with GPS verification).
- Apply for leaves or view approvals.
- Access salary slips, meeting schedules, and notices anytime.

Administrators, on the other hand, can **approve requests, assign tasks, and send notifications** directly from their phones.

The mobile application will make WORKFUSION a truly connected and user-friendly platform.

4. AI and Data Analytics:

Integrating **AI and predictive data analytics** will enhance the system's decision-making capabilities.

By analysing attendance patterns, performance trends, and payroll statistics, the system can automatically generate useful insights.

Examples include:

- Predicting employee attrition rates or absenteeism trends.
- Recommending optimal shift schedules.
- Analysing department-level performance for management reports.

This data-driven approach will help organizations make **smarter strategic decisions** and improve workforce productivity.

5. Automated Email & SMS Notifications:

To improve communication between employees and management, **automated notifications** can be integrated into the system.

The feature will automatically send **email and SMS alerts** for events such as:

- Leave application approvals or rejections.
- Meeting and task reminders.
- Salary credit confirmations or deductions.

This automation ensures that important updates are delivered instantly, enhancing coordination and reducing delays in communication.

6. Multi-Level User Roles:

Currently, the system may include basic user levels such as admin and employee. In the future, a **multi-level role-based access control system** can be implemented. New roles such as **HR Manager, Department Head, Accountant, and Project Supervisor** can be added with customized permissions.

This enhancement will:

- Enable **task delegation** and distributed responsibilities. • Ensure that users only access data relevant to their role.
- Strengthen system security and data privacy.

By defining a clear **hierarchical user structure**, large organizations can manage operations more efficiently.

7. Performance Evaluation System:

A **Performance Evaluation Module** can be introduced to automate employee assessment and recognition. The system will collect data on **attendance, task completion rate, punctuality, and project performance** to generate performance scores.

Based on this analysis, the organization can:

- Identify **high-performing employees**.
- Link performance with **rewards, promotions, or bonuses**.
- Detect underperforming areas that need improvement or training.

Such an automated evaluation system will promote **transparency, motivation, and productivity** among employees.

8. Accounting & Inventory Integration:

In the future, WORKFUSION can evolve into a **complete ERP (Enterprise Resource Planning)** platform by integrating **accounting and inventory management modules**.

This integration will allow:

- Automatic synchronization of **salary disbursements** with accounting ledgers.
- Tracking of **organizational expenses and revenues** in real time.
- Managing **inventory, supplies, and procurement** for company operations.

A unified ERP system will give management a **holistic view of financial, human resource, and operational data**, eliminating redundancy and improving efficiency.

9. Enhanced Security:

Data security remains a critical concern for any business system. Future versions of WORKFUSION will incorporate advanced protection methods such as:

- **Data encryption** for sensitive information like payroll and credentials.
- **Two-Factor Authentication (2FA)** to secure user logins.
- **Regular data backups** and secure recovery systems to prevent loss.

By strengthening security layers, the system will maintain trust, prevent unauthorized access, and ensure **data integrity** across all modules.

10. Multi-Language & Multi-Company Support:

To expand the system's usability across different organizations and regions, **multi-language and multi-company support** will be added.

This feature will allow the same platform to be customized for various **companies, branches, or departments**, each with its own set of users and configurations.

Additionally, providing support for **multiple languages** will make the system accessible to a wider user base, especially in multinational companies or diverse work environments.

16.Conclusion

The Business Management System (WORKFUSION) project has been successfully developed to automate and simplify essential business operations within an organization. It provides a centralized platform to manage employees, attendance, leaves, payroll, meetings, and announcements efficiently. By replacing manual processes with an online system, it saves time, reduces errors, and improves coordination between management and employees.

The system uses open-source technologies such as PHP, MySQL, HTML, CSS, and JavaScript, making it reliable, cost-effective, and easy to maintain. Each module was carefully designed, implemented, and tested to ensure accuracy, security, and user satisfaction. The result is a stable and user-friendly system that meets all functional and nonfunctional requirements.

Through this project, practical experience was gained in web development, database design, and system integration. It also provided a better understanding of how automation can enhance productivity and transparency in a workplace.

Although the current system meets the basic requirements of business management, it can be further enhanced in the future by adding features like mobile access, biometric attendance, AIbased analytics, and cloud deployment.

In conclusion, WORKFUSION is a complete and effective business management solution that promotes efficiency, accuracy, and transparency in daily organizational tasks. It reflects how technology can transform traditional business operations into a modern, automated, and connected environment.

17.References

1. PHP Official Documentation –
https://www.php.net/docs.php
(Used for understanding PHP syntax, server-side scripting, and backend logic development.)

2. MySQL Reference Manual –
https://dev.mysql.com/doc/
(Used for database design, relational schema creation, and SQL query optimization.)

3. W3Schools – HTML, CSS, and JavaScript Tutorials –
https://www.w3schools.com/
(For learning and implementing web development concepts, styling, and interactivity.)

4. Bootstrap Framework Documentation –
https://getbootstrap.com/docs/
(For understanding responsive web design and user interface styling.)

5. Mozilla Developer Network (MDN Web Docs) –
https://developer.mozilla.org/
(For JavaScript, CSS, and general web standards reference.)

6. XAMPP Server Setup and Configuration Guide –
https://www.apachefriends.org/download.html
(Used for local development, testing environment setup, and database connectivity.)

7. Stack Overflow Developer Community –
https://stackoverflow.com/
(Used to resolve development errors, debugging issues, and code optimization queries.)

8. GeeksforGeeks – Web Development Tutorials –
https://www.geeksforgeeks.org/web-development/
(Helpful for understanding PHP integration, form handling, and backend connectivity.)

9. TutorialsPoint – Software Engineering Concepts –
https://www.tutorialspoint.com/software_engineering/
(For theoretical concepts related to SDLC, feasibility study, and project documentation.)

10. GitHub – Open Source PHP Projects –
https://github.com/topics/php-projects
(Referred for understanding code structuring and module organization.)